SIAM J. SCI. COMPUT.
Vol. 35, No. 1, pp. B25–B42

ON THE USE OF HIGHER-ORDER PROJECTION METHODS FOR INCOMPRESSIBLE TURBULENT FLOW*

A. S. ALMGREN†, A. J. ASPDEN‡, J. B. BELL†, AND M. L. MINION§

Abstract. An important issue in the development of higher-order methods for incompressible flow is how they perform when the flow is turbulent. A useful diagnostic of a method for turbulent flow is the minimum resolution that is required to adequately resolve the turbulent energy cascade at a given Reynolds number. In this paper, we present careful numerical experiments to assess the utility of higher-order numerical methods based on this metric. We first introduce a numerical method for the incompressible Navier–Stokes equations based on fourth-order discretizations in both space and time. The method is based on an auxiliary variable formulation and combines fourth-order finite volume differencing with a semi-implicit spectral deferred correction temporal integration scheme. We also introduce, for comparison purposes, versions based on second-order spatial and/or temporal discretizations. We demonstrate that for smooth problems, each of the methods exhibits the expected order of convergence in time and space. We next examine the behavior of these schemes on prototypical turbulent flows; in particular, we consider homogeneous isotropic turbulence in which long wavelength forcing is used to maintain the overall level of turbulent intensity. We provide comparisons of the fourth-order method with the comparable second-order method as well as with a second-order semi-implicit projection method based on a shock-capturing discretization. The results demonstrate that, for a given Reynolds number, the fourth-order scheme leads to dramatic reduction in the required resolution relative to either of the second-order schemes. In addition, the resolution requirements appear to be reasonably well predicted by scaling relationships based on dimensional analysis, providing a characterization of resolution requirements as a function of Reynolds number.

Key words. higher-order projection, auxiliary formulation, spectral deferred correction

AMS subject classifications. 65M08, 76D05

DOI. 10.1137/110829386

1. Introduction.

A broad range of problems in fluid mechanics are characterized by dynamics with low Mach number. Because of the high computational cost of resolving fast time scales in these problems, many numerical methods for low Mach number flows are based on model equations derived from low Mach number asymptotics, which exploit the separation of scales between fluid motion and acoustic waves to derive specialized systems for which the natural time scale of the system is based on

*Submitted to the journal's Computational Methods in Science and Engineering section April 4, 2011; accepted for publication (in revised form) August 7, 2012; published electronically January 3, 2013. This work was supported by the Applied Mathematics Program of the DOE Office of Advanced Scientific Computing Research under the U.S. Department of Energy under contract DE-AC02-05CH11231. Simulations used resources at National Energy Research Scientific Computing Center (NERSC), which is supported by the Office of Science of the U.S. Department of Energy under contract DE-AC02-05CH11231. The U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes. Copyright is owned by SIAM to the extent not limited by these rights.
http://www.siam.org/journals/sisc/35-1/82938.html

†Center for Computational Sciences and Engineering, Lawrence Berkeley National Laboratory, Berkeley, CA 94720 (asalmgren@lbl.gov, jbbell@lbl.gov).

‡School of Engineering, University of Portsmouth, Portsmouth, Hants, PO1 3DJ, United Kingdom, and Center for Computational Sciences and Engineering, Lawrence Berkeley National Laboratory, Berkeley, CA 94720 (andrew.aspden@port.ac.uk).

§Department of Mathematics, University of North Carolina, Chapel Hill, NC 27599 (minion@email.unc.edu). This author's work was supported by the Alexander von Humboldt Foundation, and the Director, DOE Office of Science, Office of Advanced Scientific Computing Research, Office of Mathematics, Information, and Computational Sciences, Applied Mathematical Sciences Program, under contract DE-SC0004011, and the National Science Foundation under contract DMS-0854961.

the fluid velocity rather than the speed of sound. Prototypical of this type of system are the incompressible Navier–Stokes equations where passing to the zero Mach number limit of the compressible Navier–Stokes equations yields a divergence constraint on the velocity. Low Mach asymptotic model equations for more complicated systems have been developed for combustion (see, e.g., [28, 29, 33, 25, 36, 16]), atmospheric flows (see, e.g., [35, 17, 13]), and astrophysics (see, e.g., [6, 34]). Within these more general contexts, one can incorporate effects of compressibility such as those arising from reactions and other thermal processes and effects arising from stratification of the ambient background, while still formulating the problem in the context of a model that does not include acoustic wave propagation.

Many low Mach number flows of interest are turbulent. The objective of this paper is to explore the utility of higher-order discretization approaches for simulation of turbulent low Mach number flows in the simplest possible setting, namely, the incompressible Navier–Stokes equations. The convergence behavior of a higher-order algorithm for a simplified test problem is easily documented. As the resolution increases for smooth problems, higher-order methods will eventually provide more accurate solutions than lower-order methods. However, although we know that, asymptotically, a higher-order method has reduced error for smooth problems as resolution increases, we cannot make quantitative predictions of the error for different methods at a given resolution. This issue is particularly important in the context of turbulent flows because we cannot a priori assume that the minimum resolution falls within the asymptotic range of the methods. Thus, in order to quantify the potential computational advantage of using a higher-order method for turbulent flows, we would like to determine the minimum resolution required to adequately resolve turbulent flow at a given Reynolds number for discretizations with different formal order of accuracy. To do so we will use numerical experiments, focusing on forced isotropic homogeneous turbulence in which a long-wavelength forcing term is used to maintain a desired turbulence intensity, and consider the resolution requirements needed to adequately resolve both the inertial range and the dissipation range of the turbulent energy spectrum.

One common approach for incompressible flow (and more general low Mach number flows) is the use of projection-type discretizations to enforce a divergence constraint on the computed velocity. Projection methods can be thought of as fractional step schemes, wherein the equations are first evolved with a lagged approximation to the constraint, and then a projection operator is applied to push the solution back onto the divergence constraint. This type of simple fractional step scheme is inherently limited to second-order accuracy in time. Many variations of projection methods have appeared that use second-order spatial discretizations so that the overall method is second-order accurate; see, e.g., [24, 39, 5, 10], the review article [20], and Prohl [37].

Here we compare the ability of projection methods with different orders of accuracy in space and/or time to resolve turbulent flow characteristics. The fourth-order method used here is based on a variant of the auxiliary variable formulation of the Navier–Stokes equations that represents an extension to viscous flows of the fourth-order method introduced in [22]. The spatial discretization of the method is based on a finite-volume formulation, which can easily be modified to have either second- or fourth-order spatial accuracy. The temporal discretization is based on a semi-implicit spectral deferred corrections (SISDC) algorithm, which can also be trivially modified to produce either second- or fourth-order temporal accuracy. We note that there are alternative higher-order semi-implicit temporal methods that can be considered for the incompressible flow equations (e.g., [23, 7, 15, 19]). We consider here an SDC-type

approach because it provides a clear and simple way to compare second- and fourth-order methods. Although we do not consider more general low Mach number flows here, the combination of finite volume differencing and the SDC temporal integration provides a framework for temporal integration that can potentially be extended to more general low Mach number flows involving additional physical processes (see, e.g., [8]) and can be integrated with adaptive mesh refinement [2, 16, 34].

In the next section, we review the auxiliary form of the Navier–Stokes equations, and in section 3 we present the details of the numerical methods used in this study. In section 4, we first present a numerical convergence study that shows that the different SDC algorithms converge at the expected rates. We also illustrate the behavior of mixed accuracy versions that are formally second-order in space but fourth-order in time (and vice-versa). Then we consider the performance of the methods in forced homogeneous turbulence simulations. In order to show the utility of higher-order accuracy in both space and time, we compare fully fourth-order and second-order versions of the SDC algorithm and the second-order projection algorithm used in [2, 4]. We show that the fourth-order in space and time method leads to significant reduction in the minimum resolution needed to resolve the flow at a given Reynolds number. Furthermore, the resolution requirements appear to be reasonably well predicted by scaling relationships based on dimensional analysis, providing a characterization of resolution requirements as a function of Reynolds number.

2. Equations of motion. In this paper, we consider flows with vanishing Mach number and hence begin with the incompressible Navier–Stokes equations

$$\begin{aligned} \mathbf{v}_t &= -\nabla \cdot (\mathbf{v} \circ \mathbf{v} + Ip) + \nu \nabla^2 \mathbf{v} + \mathbf{H}, \\ \nabla \cdot \mathbf{v} &= 0, \end{aligned} \tag{2.1}$$

where $\mathbf{v}$ and p are the velocity and pressure, respectively; I is the identity tensor, ν is the kinematic viscosity, and $\mathbf{H}$ is an explicitly defined forcing term described in section 4. Alternative formulations of the Navier–Stokes equations can be derived by introducing a variable $\mathbf{u}^*$ that differs from the velocity by the gradient of a scalar [38, 18, 10]. Following the terminology in [22], we introduce the *auxiliary variable*, $\mathbf{u}^*$, determined by the equations

$$\mathbf{u}^*_t = -\nabla \cdot (\mathbf{u} \circ \mathbf{u} + Iq) + \nu \nabla^2 \mathbf{u}^* + \mathbf{H}, \tag{2.2}$$

$$\mathbf{u} = \mathbf{P}(\mathbf{u}^*), \tag{2.3}$$

where q is an a priori prescribed approximation to the pressure. The operator $\mathbf{P}$ is defined by $\mathbf{P}(\mathbf{u}^*) = \mathbf{u}^* - \nabla\phi$, where

$$\nabla^2 \phi = \nabla \cdot \mathbf{u}^* \tag{2.4}$$

so that $\mathbf{u} = \mathbf{P}(\mathbf{u}^*)$ is divergence-free. The explicitly defined approximation to the pressure, q, in (2.2) is equivalent to a choice of gauge in impulse methods [38]. Substituting $\mathbf{u}^* = \mathbf{u} + \nabla\phi$ into (2.2) confirms that $\mathbf{u}$ satisfies (2.1) with

$$p = q + \phi_t - \nu \nabla^2 \phi. \tag{2.5}$$

Since (2.5) implies $p - q = \phi_t - \nu\nabla^2\phi$, the closer q is to the exact pressure, p, the closer the auxiliary variable, $\mathbf{u}^*$, is to the exact velocity, $\mathbf{u}$. Hence in our numer-

ical testing, q is reset to approximate the pressure, p, at the beginning of each time step, and held fixed over the time step. Since in this study only periodic boundary conditions are considered and an "exact" projection operator is being used (see section 3.2.2), the choice of q does not change the accuracy of the method. Additional detail on how (2.5) is used to update the pressure in the numerical method is included at the end of section 3.4.

The advantage of the auxiliary variable approach is that $\mathbf{u}^*$ is not subject to a divergence constraint, hence a higher-order temporal discretization can be applied directly to the evolution equation for $\mathbf{u}^*$. In the following section, we present a fourth-order discretization of the auxiliary variable equations using a conservative finite-volume method in space and a deferred correction method in time.

We have omitted a discussion of boundary conditions for the equations of motion. The focus of this paper is on evaluating the benefits of a higher-order discretization for bulk flow phenomena and hence the numerical tests are done in simplified periodic geometries. The construction of higher-order temporal methods for PDEs with time-dependent boundary conditions (even for simple equations) is not straightforward, and many papers devoted to avoiding a reduction of order at the boundary have appeared [21, 12, 1, 3, 11]. The best treatment of boundary conditions for higher-order, semi-implicit methods for divergence constrained flows is still an open research problem.

3. Method. In [22], a fourth-order (in time and space) method for the constant and variable density, inviscid, low Mach number asymptotic equations in two dimensions is presented. The numerical method used in this study is an extension of the method in [22] to viscous flows in three dimensions. The main modification to the method in [22] is the use of a semi-implicit spectral deferred corrections (SISDC) method to treat the diffusive terms. Coupling of the SISDC method with an auxiliary variable formulation has appeared in [30, 32]. In principle, other semi-implicit temporal schemes could be used instead; however, there are two main motivations for the use of the SDC method. The first is that the order of the method is easily determined by specifying the number of deferred correction iterations. The second is that the particular numerical implementation used for the numerical studies is designed for use on problems with more complicated equations where multiple operator splitting and multirate time integration is desirable (as in, e.g., [8, 26, 9]).

3.1. Finite volume formulation. To facilitate the explanation, several notational conventions are first introduced. We assume that the three-dimensional domain is divided into a uniform array of cells of length, width, and height h. Let the cell with center at (x_i, y_j, z_k) be denoted by $V_{i,j,k}$, and let the half-integer subscripts $i + 1/2$, $j+1/2$, $k+1/2$ denote a shift by distance $h/2$ in the x-, y-, and z-direction, respectively. We also denote by $E_{i+1/2,j,k}$ the face of $V_{i,j,k}$ corresponding to $x_{i+1/2,j,k} = x_i + h/2$; i.e., $E_{i+1/2,j,k} = \{x_{i+1/2}\} \times [y_{j-1/2}, y_{j+1/2}] \times [z_{k-1/2}, z_{k+1/2}]$. The other faces are defined analogously.

The finite-volume approach is based on an evolution equation for the cell average of the auxiliary variable $\mathbf{u}^*$ defined by

$$\bar{\mathbf{u}}^*_{i,j,k}(t) = \frac{1}{h^3} \int_{V_{i,j,k}} \mathbf{u}^*(x, y, z, t)\, dx\, dy\, dz. \tag{3.1}$$

The finite-volume discretization updates cell averages by the construction of fluxes

that are defined as averages over the faces of the cells. For example,

$$\tilde{f}(t)_{i+1/2j,k} = \frac{1}{h^2}\int_{E_{i+1/2,j,k}} f(x_{i+1/2}, y, z, t)\, dz\, dy \tag{3.2}$$

with the analogous formulae for other faces.

As a further notational convenience, we also use a tilde without index shifting when referring to the cell-edge averages of a vector quantity when the first component of the vector is averaged over $E_{i+1/2,j,k}$, the second component is averaged over $E_{i,j+1/2,k}$, and the third over $E_{i,j,k+1/2}$. This convention will also be followed for gradients at faces, hence, for example,

$$\tilde{\nabla}\phi_{i,j,k} = ((\tilde{\phi}_x)_{i+1/2,j,k}, (\tilde{\phi}_y)_{i,j+1/2,k}, (\tilde{\phi}_z)_{i,j,k+1/2}). \tag{3.3}$$

To specify the finite-volume formulation of the conservation law

$$Q_t + \nabla \cdot F(Q) = S, \tag{3.4}$$

where $Q(x, y, z, t)$ is the vector of conserved quantities and $F(Q) = (f_1(Q), f_2(Q), f_3(Q))$ is the flux function, we integrate the equation over a computational cell and use the divergence theorem to attain

$$\frac{d}{dt}\bar{Q}(t)_{i,j,k} + \frac{1}{h^3}\int_{\partial V_{i,j,k}} F(Q(x, y, z, t)) = \bar{S}(t)_{i,j,k}. \tag{3.5}$$

In this equation, the flux integral is defined as

$$\begin{aligned}
&\int_{\partial V_{i,j,k}} F(Q(x, y, z, t))\, dx\, dy\, dz \\
&\quad = \int_{E_{i+1/2,j,k}} f_1(Q(x_{i+1/2}, y, z, t))\, dz\, dy - \int_{E_{i-1/2,j,k}} f_1(Q(x_{i-1/2}, y, z, t))\, dz\, dy \\
&\quad + \int_{E_{i,j+1/2,k}} f_2(Q(x, y_{j+1/2}, z, t))\, dz\, dx - \int_{E_{i,j-1/2,k}} f_2(Q(x, y_{j-1/2}, z, t))\, dz\, dx \\
&\quad + \int_{E_{i,j,k+1/2}} f_3(Q(x, y, z_{k+1/2}, t))\, dy\, dx - \int_{E_{i,j,k-1/2}} f_3(Q(x, y, z_{k-1/2}, t))\, dy\, dx,
\end{aligned}$$

or using the definition of face average

$$\begin{aligned}
\frac{1}{h^3}\int_{\partial V_{i,j,k}} F(Q(x, y, z, t)) &= \frac{\tilde{f}_1(Q(t))_{i+1/2,j,k} - \tilde{f}_1(Q(t))_{i-1/2,j,k}}{h} \\
&+ \frac{\tilde{f}_2(Q(t))_{i,j+1/2,k} - \tilde{f}_2(Q(t))_{i,j-1/2,k}}{h} \\
&+ \frac{\tilde{f}_3(Q(t))_{i,j,k-1/2} - \tilde{f}_3(Q(t))_{i,j,k-1/2}}{h}.
\end{aligned} \tag{3.6}$$

Since the right-hand side of this equation resembles a discretized divergence, we also write

$$\frac{1}{h^3}\int_{\partial V_{i,j,k}} F(Q(x, y, z, t)) = \tilde{\nabla} \cdot \tilde{F}(Q)_{i,j,k},$$

i.e., the operator $(\tilde{\nabla}\cdot)$ is the sum of simple differences of averaged quantities over faces.

Applying the above definitions to (2.2) yields the system of ODEs

$$\frac{d}{dt}\bar{\mathbf{u}}^*(t)_{i,j,k} = -\tilde{\nabla}\cdot\tilde{F}(\mathbf{u}^*,q)_{i,j,k} + \bar{\mathbf{H}}(t)_{i,j,k}, \tag{3.7}$$

where we treat q as known and consider $\mathbf{u}$ to be computable from $\mathbf{u}^*$ using (2.3). Note that (3.7) is mathematically exact, i.e., no numerical approximations have been introduced up to this point. The flux function, F, in (3.7) is split into two pieces,

$$F(\mathbf{u}^*,q) = A(\mathbf{u}^*,q) + D(\mathbf{u}^*), \tag{3.8}$$

where A (which contains the nonlinear terms in F) is treated explicitly in the temporal integration scheme and D (the diffusive terms) is treated implicitly. The discretization of A is described in detail next, followed by the details of the temporal integration method.

3.2. Explicit discretization of nonlinear terms. The explicit part of the flux function in (3.8), $A(\mathbf{u}^*,q) = (a_1(\mathbf{u}^*,q), a_2(\mathbf{u}^*,q), a_3(\mathbf{u}^*,q))$, is defined as

$$a_1 = \begin{pmatrix} uu+q \\ uv \\ uw \end{pmatrix}, \quad a_2 = \begin{pmatrix} vu \\ vv+q \\ vw \end{pmatrix}, \quad a_3 = \begin{pmatrix} wu \\ wv \\ ww+q \end{pmatrix}. \tag{3.9}$$

Here $\mathbf{u} = (u,v,w)$ is defined by (2.3) and the discretization of the projection, $\mathbf{P}$, is described in section 3.2.2.

To compute the nonlinear term, $\tilde{\nabla}\cdot\tilde{F}(\mathbf{u}^*,q)$, in (3.7), it is necessary to construct an accurate approximation of averages of the flux function, namely $\tilde{A}(\mathbf{u}^*,q)$, from cell average quantities $\bar{\mathbf{u}}^*$ (and $\bar{q}$). In the temporal method, $\tilde{A}(\mathbf{u}^*,q)$ is treated explicitly, and we denote the approximation at a given time t_m by $\tilde{A}(\bar{\mathbf{u}}^{*,m},\bar{q}^m)$. The computation of $\tilde{A}(\bar{\mathbf{u}}^{*,m},\bar{q}^m)$ proceeds in three separate steps:

1. computing the averages over faces of $\tilde{\mathbf{u}}^{*,m}$ and $\tilde{q}^m$ from the cell averages $\bar{\mathbf{u}}^{*,m}$ and $\bar{q}^m$,
2. applying a projection operator to $\tilde{\mathbf{u}}^{*,m}$ to yield divergence-free face averages $\tilde{\mathbf{u}}^m$,
3. computing the averages of the flux function $\tilde{A}(\bar{\mathbf{u}}^{*,m},\bar{q}^m)$ from the averages $\tilde{\mathbf{u}}^m$ and $\tilde{q}^m$.

3.2.1. Computing averages on faces. Given cell average values, $\bar{\phi}_{i,j,k}$, a fourth-order approximation to the average of ϕ over face $E_{i+1/2,j,k}$ is

$$\tilde{\phi}_{i+1/2,j,k} = \frac{-\bar{\phi}_{i-1,j,k} + 7(\bar{\phi}_{i,j,k} + \bar{\phi}_{i+1,j,k}) - \bar{\phi}_{i+2,j,k}}{12}. \tag{3.10}$$

This approximation is derived by simply integrating a standard one-dimensional interpolation formula over the face. Equation (3.10) is applied to $\bar{q}$ and the components of $\bar{\mathbf{u}}^*$ that are normal to each face.

In finite-volume methods for hyperbolic problems, limiters are often applied to the formula given in (3.10) near sharp gradients in the solution to avoid introducing oscillations in the numerical solution (see, e.g., [14]) when the solution is not well resolved. A similar procedure has also been employed in an SDC-based method for

one-dimensional problems in [26]. Here, no limiters are used since our focus is on understanding the behavior of the method when the solution is well resolved, not on increasing the robustness of the method when the solution is underresolved. For second-order versions of the method (3.10) is replaced by a simple average.

3.2.2. The numerical projection. The fluxes defined in (3.9) contain the averages over faces of the divergence-free velocity, $\mathbf{u} = (u, v, w)$. Hence, before the averages of fluxes over faces can be computed, the face average of $\tilde{\mathbf{u}}$ must be computed from those of $\tilde{\mathbf{u}}^*$ through a numerical projection.

The divergence-free velocities, $\tilde{\mathbf{u}}$, are computed by solving a discrete version of (2.4) averaged over cells,

$$\tilde{\nabla} \cdot \tilde{\nabla}_h \bar{\phi}_{i,j,k} = \tilde{\nabla} \cdot \tilde{\mathbf{u}}^*_{i,j,k}. \tag{3.11}$$

Equation (3.11) is solved for an approximation to $\bar{\phi}_{i,j,k}$ where face averages of $\tilde{\mathbf{u}}^*$ are approximated by the analogue of (3.10), and the averages of normal derivatives at faces $\tilde{\nabla}\phi_{i,j,k}$ (see (3.3)) are approximated by a fourth-order centered formula, $\tilde{\nabla}_h \bar{\phi}_{i,j,k}$, which, e.g., at $E_{i+1/2,j,k}$ is

$$(\tilde{\phi}_x)_{i+1/2,j,k} = \frac{\bar{\phi}_{i-1,j,k} + 15(-\bar{\phi}_{i,j,k} + \bar{\phi}_{i+1,j,k}) - \bar{\phi}_{i+2,j,k}}{12h}. \tag{3.12}$$

This yields a 13-point stencil for the discrete Laplacian operator $\tilde{\nabla} \cdot \tilde{\nabla}_h$. The resulting linear system is solved using a standard multigrid procedure. For methods with second-order spatial accuracy, $\tilde{\mathbf{u}}^*$ is again computed with a simple average rather than (3.10), and (3.12) becomes a two-point centered difference that yields the standard 7-point stencil for $\tilde{\nabla} \cdot \tilde{\nabla}_h$. Then

$$\tilde{\mathbf{u}}_{i,j,k} = \tilde{\mathbf{u}}^*_{i,j,k} - \tilde{\nabla}_h \phi_{i,j,k} \tag{3.13}$$

are divergence-free edge averages in the sense that

$$\tilde{\nabla} \cdot \tilde{\mathbf{u}}_{i,j,k} = 0. \tag{3.14}$$

In the parlance of projection methods, we are using an "exact" projection of the values $\tilde{\mathbf{u}}^*$, i.e., the averages over faces of the normal velocity components of $\tilde{\mathbf{u}}$ satisfy a discrete divergence constraint up to the accuracy of the elliptic solver. However, in the flux functions defined in (3.9), all three components of the divergence-free velocity, $\mathbf{u} = (u, v, w)$, are required at each face. The projection procedure just described determines only the normal velocity at each face. The additional tangential velocities are derived from the solution of (3.13) by first computing cell-average velocities,

$$\bar{\mathbf{u}}_{i,j,k} = \bar{\mathbf{u}}^*_{i,j,k} - \bar{\nabla}_h \bar{\phi}_{i,j,k}, \tag{3.15}$$

where the average of the gradient, $\bar{\nabla}_h$, is computed using a centered difference formula applied to $\bar{\phi}_{i,j,k}$ (the solution of (3.11)), e.g.,

$$(\bar{\phi}_x)_{i,j,k} = \frac{\bar{\phi}_{i-2,j,k} + 8(-\bar{\phi}_{i-1,j,k} + \bar{\phi}_{i+1,j,k}) - \bar{\phi}_{i+2,j,k}}{12h}. \tag{3.16}$$

Averages of the tangential velocities on faces are then computed by using (3.10) (or the second-order analog) on the appropriate components of $\bar{\mathbf{u}}_{i,j,k}$. Neither $\bar{\mathbf{u}}_{i,j,k}$ nor the average of tangential velocities on faces satisfy a discrete divergence constraint.

3.2.3. Computing nonlinear terms. In order to compute higher-order accurate values of averages of the flux functions, it is necessary to compute the average over faces of the products of velocities appearing in (3.9). The primary difficulty in building higher-order finite volume methods is that the average of a product is not equal to the product of averages. We proceed as in [22] by expressing averages of a product as the product of averages plus a correction term that depends on approximations to the tangential derivatives of the quantities on the face. To achieve fourth-order accuracy, it is sufficient to include only the first derivatives in the correction.

For example, for an arbitrary quantity $\tilde{\phi}$ on face $E_{i+\frac{1}{2},j,k}$,

$$(\widetilde{\phi\rho})_{i+1/2,j,k} = (\tilde{\phi}_{i+1/2,j,k})(\tilde{\rho}_{i+1/2,j,k}) + \frac{h^2}{12}(\tilde{\phi}_y\tilde{\rho}_y + \tilde{\phi}_z\tilde{\rho}_z) + O(h^4), \tag{3.17}$$

where, for example,

$$\tilde{\phi}_y = \frac{-5\tilde{\phi}_{i+1/2,j+2,k} + 34(\tilde{\phi}_{i+1/2,j+1,k} - \tilde{\phi}_{i+1/2,j-1,k}) + 5\tilde{\phi}_{i+1/2,j-2,k}}{48h}. \tag{3.18}$$

When a second-order finite-volume spatial discretization is used, (3.10) is replaced by a simple average of adjacent cells, and the $O(h^2)$ correction terms in (3.17) are omitted. Once the appropriate averages of products of velocities at each cell face have been computed, $\tilde{A}(\tilde{\mathbf{u}}^*)$ is computed by the simple difference given in (3.6).

3.3. Discretization of diffusive terms. To complete the description of the spatial discretization, we now describe the computation of the diffusive terms, $\tilde{\nabla} \cdot D(\mathbf{u}^*)$. From (2.2) and the definitions above,

$$\tilde{\nabla} \cdot D(\mathbf{u}^*) = \nu\tilde{\nabla} \cdot \tilde{\nabla}\mathbf{u}^*. \tag{3.19}$$

Given cell averages of $\bar{\mathbf{u}}^{*,m}$ at t_m, we approximate this term using (3.12),

$$\tilde{\nabla} \cdot \tilde{D}(\bar{\mathbf{u}}^{*,m}) = \nu\tilde{\nabla} \cdot \tilde{\nabla}_h\bar{\mathbf{u}}^{*,m}. \tag{3.20}$$

In the time-stepping method described in the next section, the diffusive terms are computed implicitly except at the beginning of each time step where (3.20) is used explicitly.

3.4. Temporal discretization. The spatial discretization described in the previous section is integrated in time using a method of lines approach based on a semi-implicit spectral deferred corrections (SISDC) method [31]. The basic approach in the SDC method is to advance the solution of the ODE from time t_n to t_{n+1} through the use of intermediate values defined by nodes in the interval $[t_n, t_{n+1}]$, which here are denoted generically t_m. The SDC method proceeds by first computing a provisional solution using a first-order forward/backward Euler step at each of the nodes t_m. Then, a series of corrections sweeps are performed wherein an approximation to the error or correction to the provisional solution is computed by a similar first-order method at the nodes t_m. In each of the corrections sweeps, the equation for the correction contains an explicitly computed approximation to the temporal integral of the right-hand side of the ODE, which is computed using a quadrature rule applied to the values at the nodes t_m. For specific details of semi-implicit SDC methods, the reader is referred to [31, 27]. For an example of SDC applied to projection methods, see [30, 32].

Each correction sweep of SDC raises the formal order of accuracy of the overall method by one when a first-order forward/backward Euler approximation scheme is used. Hence for the fourth-order temporal methods, four total SDC sweeps (including the provisional sweep) are performed, while for the second-order methods, two sweeps are done. The maximum formal order of SDC methods is that of the underlying quadrature rule defined on the nodes t_m. Here, for the fourth-order methods we use three Gauss-Lobatto nodes in the SDC sweeps (including the endpoints t_n and t_{n+1}) so that the quadrature rule is equivalent to Simpson's rule. For second-order temporal accuracy, the quadrature rule is simply the trapezoid rule, hence no intermediate nodes are actually used. A detailed study of the the choice of quadrature nodes for semi-implicit SDC methods appears in [27].

Here we provide a concise summary of one semi-implicit substep in the SDC time integration method for updating cell average values $\bar{\mathbf{u}}^*$. Superscripts are used to denote the time level of each approximation; for example, the approximation to the cell-averaged value $\bar{\mathbf{u}}^*(x_i, y_j, z_k, t_m)$ is denoted $\bar{\mathbf{u}}^{*,m}_{i,j,k}$. Superscripts or subscripts are suppressed when the meaning is apparent.

At the beginning of each time step, we have the quantities $\bar{\mathbf{u}}^{*,n}$ and $\bar{p}^n$. Before the SDC sweeps are begun, $\tilde{\nabla} \cdot (\tilde{A}(\bar{\mathbf{u}}^{*,n}, \bar{q}^n) + \tilde{D}(\bar{\mathbf{u}}^{*,n}))$ and $\bar{\mathbf{H}}^n$ are computed. The value of $\bar{q}^m$ is also set to $\bar{p}^n$ for each m.

Next a provisional solution is approximated at each substep defined by t_{m+1} using the forward/backward Euler update

$$(I - \nu \Delta t_m \tilde{\nabla} \cdot \tilde{D}) \bar{\mathbf{u}}^{*,m+1} = \bar{\mathbf{u}}^{*,m} + \Delta t_m (-\tilde{\nabla} \cdot \tilde{A}(\bar{\mathbf{u}}^{*,m}, \bar{q}^m) + \bar{\mathbf{H}}^m), \tag{3.21}$$

where $\Delta t_m = t_{m+1} - t_m$. This equation is solved with a standard multigrid method. After $\bar{\mathbf{u}}^{*,m+1}$ is computed, new values $\tilde{\nabla} \cdot \tilde{A}(\bar{\mathbf{u}}^{*,m+1})$ are computed for the next substep.

Once the provisional solution is computed, additional SDC correction sweeps are done to improve the accuracy of the provisional solution (three sweeps for the fourth-order temporal accuracy and one for the second order). In each correction sweeps k, a similar first-order semi-implicit method is done at each substep to a modified equation of the form

$$\begin{aligned} &(I - \nu \Delta t_m \tilde{\nabla} \cdot \tilde{D}) \bar{\mathbf{u}}^{*,m+1,k+1} = \bar{\mathbf{u}}^{*,m,k+1} \\ &+ \ \Delta t_m \left(\tilde{\nabla} \cdot (-\tilde{A}(\bar{\mathbf{u}}^{*,m,k+1}, \bar{q}^n) + \tilde{A}(\bar{\mathbf{u}}^{*,m,k}, \bar{q}^n)) \right) \\ &- \ \nu \Delta t_m \tilde{\nabla} \tilde{D}(\bar{\mathbf{u}}^{*,m,k}) + \mathbf{I}_m^{m+1}(\bar{\mathbf{u}}^{*,m,k}, \bar{q}^n), \end{aligned} \tag{3.22}$$

where the term

$$\mathbf{I}_m^{m+1}(\bar{\mathbf{u}}^{*,m,k}, \bar{q}^n) \approx \int_{t_m}^{t_{m+1}} \tilde{\nabla} \cdot \left(-\tilde{A}(\bar{\mathbf{u}}^{*,k}, \bar{q}^n) + \nu \tilde{D}(\bar{\mathbf{u}}^{*,k}) \right) + \bar{\mathbf{H}} \, dt. \tag{3.23}$$

See [31] for a more detailed derivation of the correction equation.

Finally, at the end of each full time step, two additional tasks are completed:

(1) Reset $\bar{\mathbf{u}}^{*,n+1}$ to $\bar{\mathbf{u}}^{n+1}$, which has been computed in the projection step of the computation of $\tilde{A}(\bar{\mathbf{u}}^{*,m+1}, \bar{q}^n)$ in the final SDC substep as discussed in section 3.2.2.

(2) Compute an update to $\bar{p}^{n+1}$ by discretizing (2.5),

$$\bar{p}^{n+1} = \bar{q}^n + \frac{\bar{\phi}^{n+1} - \bar{\phi}^n}{\Delta t} - \nu \tilde{\nabla} \cdot \tilde{\nabla}_h \bar{\phi}^{n+1}. \tag{3.24}$$

A rigorous analysis of the discretization approach developed here for incompressible Navier–Stokes is not feasible. However, we have performed a linear stability analysis of the integration schemes proposed here for a linear advection/diffusion problem. Results of that analysis are presented in the appendix.

This pressure update yields only a temporally second-order update of the pressure, but this does not affect the accuracy of the velocities. If an accurate value of pressure were desired, the time derivative term in (3.24) equation must be higher-order accurate. In [22] this derivative is computed to fourth-order accuracy by using the cell average values of ϕ at five SDC substeps. Here, the numerical diagnostics do not include the pressure, hence, the lower-order update is sufficient.

For the results presented here, $\bar{p}$ and $\bar{q}$ are set to zero at the initial time step. For each subsequent time step $\bar{q}^n$ is initialized to the approximation of $\bar{p}^{n+1}$ given by (3.24) from the previous time step, and is then held constant for each SDC substep.

4. Numerical results. In this section we first demonstrate the convergence behavior of the SDC schemes for smooth problems. For this first series of tests we consider four variants of the SDC algorithm, S2T2, S2T4, S4T2, S4T4, where for SnTm, n refers to the spatial order and m refers to the temporal order. For the second-order temporal discretization, the quadrature uses the trapezoidal rule and a single SDC iteration is required. For the fourth-order temporal discretization, we use Simpson's rule for the quadrature with three SDC iterations.

For the second example, we investigate in more detail the performance of S2T2 and S4T4 on the simulation of three-dimensional maintained homogeneous isotropic turbulence. Both tests are based on the same basic configuration in which we specify a smooth initial velocity profile and a smooth forcing term at large scales. In particular the initial conditions are a single Fourier mode (to give a sensible estimate for Δt). The turbulence is maintained through a time-dependent zero-mean source term in the momentum equation consisting of a superposition of long-wavelength Fourier modes, following [4]. Specifically, the domain is a triply periodic unit cube, and the forcing term in (2.1) for each example is specified to be

$$\mathbf{H}(\mathbf{x},t) = \sum_{|\boldsymbol{\kappa}|\in[1,4]} \mathbf{a}_{i,j,k} \cos\left(f_{i,j,k}t + \omega_{i,j,k}\right) \cos(2\pi\kappa_i x + \psi_{i,j,k}) \times \cos\left(2\pi\kappa_j y + \eta_{i,j,k}\right) \cos(2\pi\kappa_k z + \zeta_{i,j,k})$$

for random amplitudes $\mathbf{a}_{i,j,k}$, frequencies $f_{i,j,k} \in [\pi, 2\pi)$, and phases $\omega_{i,j,k}$, $\psi_{i,j,k}$, $\eta_{i,j,k}$, and $\zeta_{i,j,k} \in [0, 2\pi)$. The early time behavior of this system, before the turbulent cascade has had time to populate the higher frequencies, provides a canonical example of a smooth flow problem. At later times this system transitions to fully developed turbulent flow, which we use to examine the behavior for turbulence simulations.

4.1. Convergence tests. For the convergence tests we consider five different resolutions ranging from 32^3 to 512^3 and compare the relative errors. For these simulations we set the kinematic viscosity, $\nu = 2.0 \times 10^{-3}$, so that the solution is well resolved even on the coarsest grid. In each case we choose a Δt proportional to Δx, and hold that value fixed for the simulation. We consider three different ratios of Δt to Δx. The largest, corresponding to $\Delta t_L = 0.012$ for the 32^3 grid, is based on approximating the maximum stable time step for S2T2. We also consider two smaller time steps corresponding to a reduction of Δt_L by factors of 2 and 4, respectively. The final time is $T = 0.12$ for the runs with the large time step, $T = 0.06$ for the medium time step, and $T = 0.03$ for the smallest time step. We consider all five different resolutions

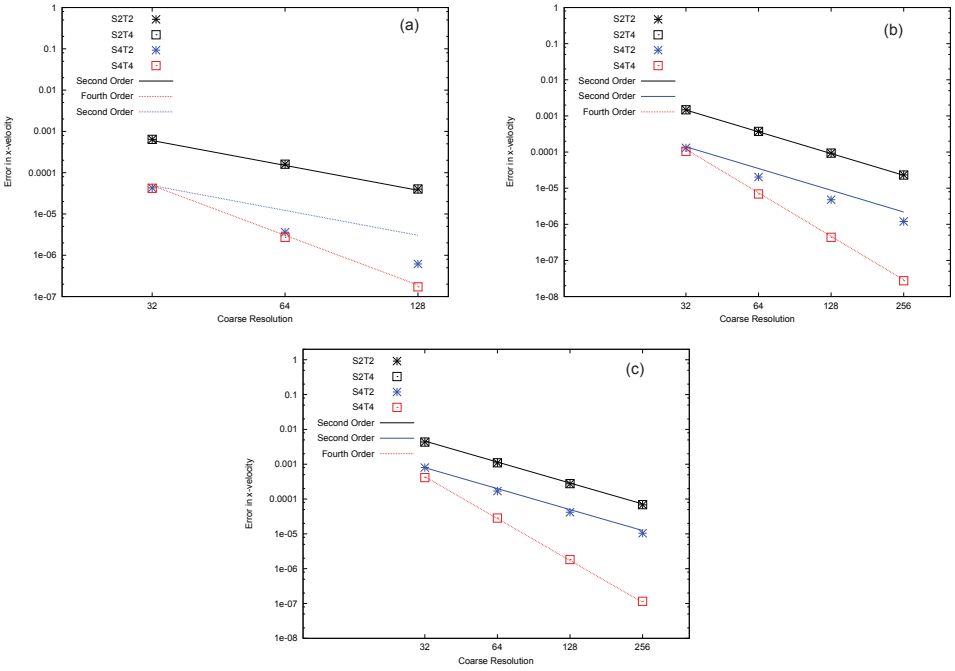

FIG. 4.1. *Here we show the convergence behavior of the S2T2, S2T4, S4T2, and S4T4 algorithms. The upper two lines in each panel show ideal second-order convergence; the lowest line in each shows ideal fourth-order convergence. Panel* (a) *shows results using the small time step; panel* (b) *shows results using the medium time step and panel* (c) *shows results using the largest stable time step for these methods. We note that for each panel the number of time steps taken for the coarsest mesh is the same; thus, in each panel the final times are different. We note that the S2T2 and S2T4 data show almost ideal second-order behavior and the S4T4 data shows almost ideal fourth-order behavior. The S4T2 data show intermediate convergence.*

at the largest and medium time step; for the smallest time step, we consider only the four resolutions as the trends are already apparent. Convergence behavior in L^2, estimated by comparing solutions at adjacent resolutions for the small, medium, and large values of Δt are presented in Figure 4.1. For the smooth flow problem considered here, essentially the same results are obtained in L^1 and L^∞. For the smallest time step, the error for the spatially second-order methods is dominated by the spatial error; e.g., S2T2 and S2T4 show second-order behavior with essentially no improvement from the fourth-order temporal differencing in S2T4. S4T4 shows consistent fourth-order convergence. S4T2 initially shows fourth-order behavior; however, with more refinement, the reduced temporal accuracy begins to dominate the error. At the medium time step, S2T2 and S2T4 remain second-order and S4T4 remains fourth-order. For the medium time step, S4T2 is initially somewhat better than second-order but at higher resolutions the convergence reduces to second-order. At the largest time step, S4T4 remains fourth-order accurate while all of the other variants now exhibit second-order convergence. The higher-order spatial treatment in S4T2 improves the overall accuracy but does not alter the rate of convergence.

4.2. Turbulent flow diagnostics. We now consider the performance of the SDC approach when applied to a more complex flow, i.e., maintained homogeneous isotropic turbulence. The goal here is to assess the potential advantage of using a

fourth-order rather than second-order discretization when the motivation is to be able to use the coarsest possible spatial resolution that can still accurately resolve the turbulent flow. For these tests, we restrict consideration to S2T2 and S4T4. In addition, we also include a comparison to the methodology used for the adaptive incompressible flow solved discussed in [2]. The advective discretizations in this approach are based on unsplit second-order Godunov type methodology adapted from shock-capturing schemes. Here we consider a piecewise linear version of the algorithm, denoted IAMR, and a piecewise parabolic version, denoted PPM.

As noted before, we use the same basic configuration as was used for the convergence tests. To enable a more detailed comparison, we run the S4T4 algorithm at a resolution of 256^3 until the flow has transitioned to a well-developed turbulent flow. We then restart each of the methods with coarsened versions of this data and run for approximately one eddy turnover time. We also continue the 256^3 S4T4 simulation to the same time as the coarser versions. This run will be referred to as the high-resolution solution hereafter.

For the first case, we set $\nu = 3 \times 10^{-4}$, corresponding to a peak Taylor Reynolds number of $\mathrm{Re}_\lambda = u\lambda/\nu \approx 62$, where the Taylor microscale is defined as $\lambda^2 = 15\nu u^2/\varepsilon$, the energy dissipation rate is $\varepsilon = u^3/l$ for integral length scale $l \approx 0.1$ and rms velocity fluctuation $u \approx 0.775$ (arbitrary units). Simulation results of the magnitude of vorticity for each of the four methods on a 128^3 grid are presented in Figure 4.2 along with the high-resolution simulation. The (x, z)-plane at $y = 0$ is shown, but this choice is arbitrary given that the boundary conditions are periodic. All data are taken from exactly the same point in time, which corresponds to a local peak in the kinetic energy. We note that the shock-capturing schemes both produce reasonable looking solutions but are somewhat lacking in fine scale detail compared to the high-resolution simulation. The second-order S2T2 scheme, on the other hand, appears to have more fine-scale detail than the high-resolution simulation. The S4T4 scheme at 128^3, not surprisingly, appears to be closer to the high-resolution simulation than the other approaches. To make this comparison more precise, we plot in Figure 4.3 the compensated spectrum from the simulations. The compensated spectrum is given by $\kappa^{5/3}E(\kappa)$ (evaluated pointwise) where $E(\kappa)$ is the standard energy spectrum. It is obtained by computing the FFT of the velocity components, computing the energy in κ space and multiplying by the $\kappa^{5/3}$. The spectrum is then computed by averaging over spherical shells of width one. In the compensated spectrum a $\kappa^{-5/3}$ inertial range appears flat. We note that the data has not been normalized in any way. From Figure 4.3 we can see that the shock-capturing schemes have significantly less energy at higher wavelengths than the high-resolution solution. This reflects the role of numerical dissipation in these schemes and is consistent with the loss of fine-scale detail in Figure 4.2. The second-order SDC scheme, on the other hand, does not dissipate enough energy at higher wave lengths so the spectrum lies above the high-resolution solution for high κ. This difference in behavior when the flow is underresolved is a consequence of the centered treatment of advection in S2T2 versus the upwinding approach used in IAMR and PPM. Finally, S4T4 does a good job of tracking the spectrum of the high-resolution solution. We note that the Reynolds number considered here is close to the largest value that can be resolved with S4T4 on a 128^3 grid; none of the schemes provide an acceptable solution at 64^3. In Figure 4.4, we provide further comparison, showing the compensated spectrum for the second-order schemes at 256^3 compared to S4T4 at 128^3. The results here are roughly comparable, suggesting that the use of the fourth-order SDC algorithm reduces the

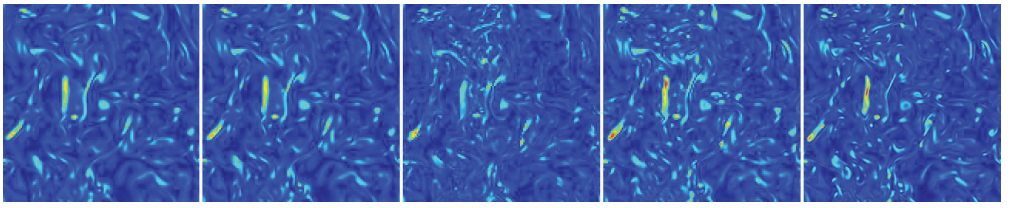

FIG. 4.2. *Simulations for high Reynolds number case. Each panel represents a contour plot of the magnitude of vorticity on an (x, z)-plane corresponding to $y = 0$.*

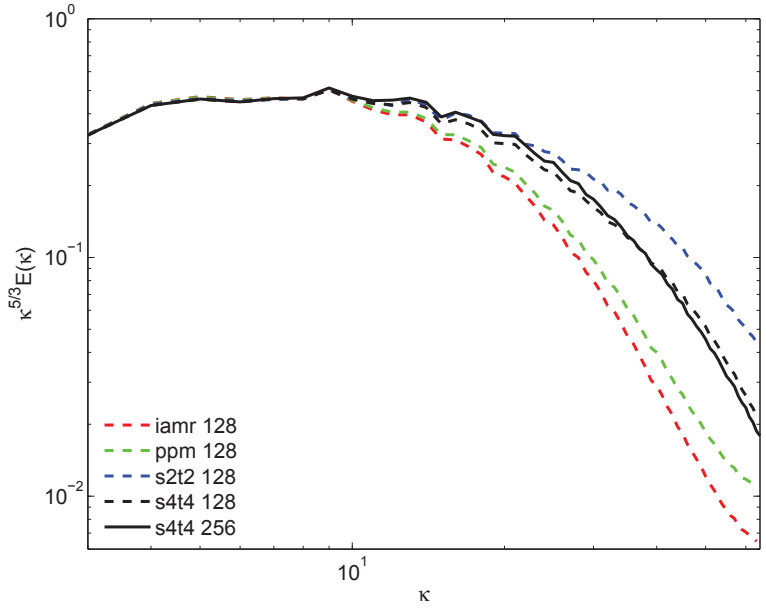

FIG. 4.3. *Compensated spectrum for high Reynolds number case comparing schemes at* 128^3 *resolution.*

computational requirements by about a factor of two in each spatial dimension, thus reducing the total number of points (space × time) advanced to reach a specified time by a factor of 16 for a three-dimensional simulation.

To assess the impact of the higher-order algorithm on execution time, we measured both S2T2 and S4T4 on a 128^3 grid. For this test case the second-order algorithm used 1570 seconds whereas the fourth-order algorithm used 4405 seconds. Thus, the higher-order algorithm increased the computational cost per grid point by a factor of 2.8. Given that the higher-order algorithm reduces the number of zones advanced by a factor of 16, the net computational advantage of the higher-order algorithm is approximately a factor of 5.7.

We would expect that dimensional analysis would enable us to scale resolution requirements with Reynolds number. In particular, the minimum resolution needed to adequately resolve a turbulent flame should scale with the Kolmogorov length scale, η. We can estimate $\eta = (\nu^3/\varepsilon)^{1/4}$, which allows us to rewrite the Taylor Reynolds number as $\mathrm{Re}_\lambda \sim (l/\eta)^{2/3}$. A reasonable assumption is that the relationship

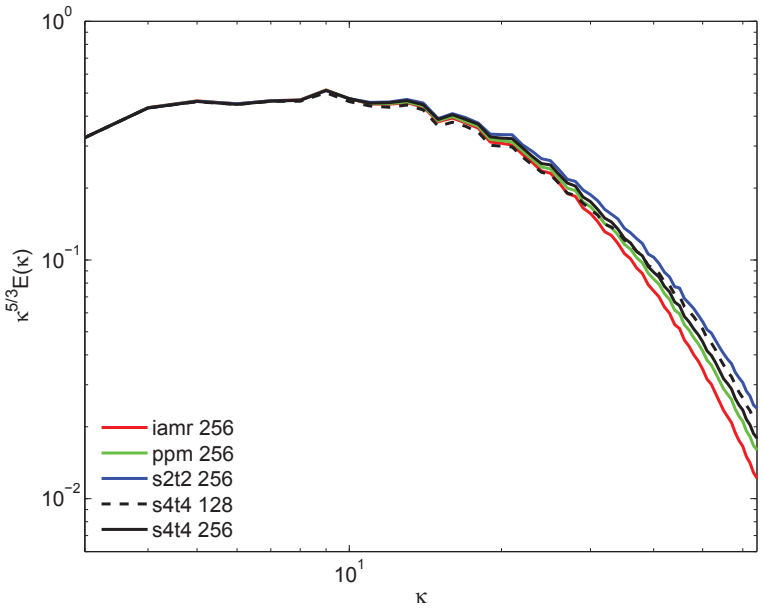

FIG. 4.4. *Compensated spectrum for high Reynolds number case comparing schemes at* 256^3 *to S4T4 at* 128^3.

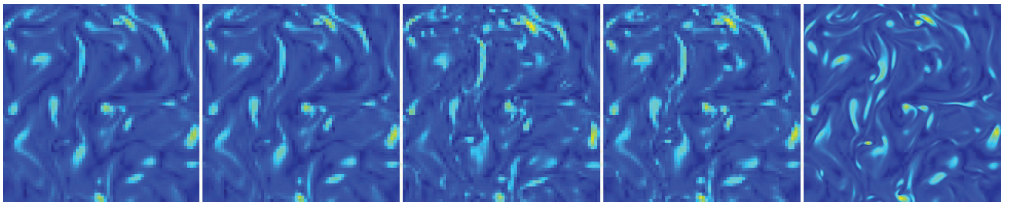

FIG. 4.5. *Simulations for low Reynolds number case. Each panel represents a contour plot of the magnitude of vorticity on an* (x, z)*-plane corresponding to* $y = 0$.

between resolution and the Kolmogorov scale is linear; i.e., the minimum Δx is a constant multiple of η where the constant is a property of the particular method. With this assumption, since l is a large-scale property of the flow, we can estimate the resolution needed for a given Re_λ for a given method in terms of this constant. Based on this analysis, we would predict that the maximum Reynolds number that we could resolve with S4T4 at 64^3 would be $\mathrm{Re}_\lambda \approx 39$, corresponding to viscosity, $\nu = 7.6 \times 10^{-4}$. In Figure 4.5 we present vorticity slices from 64^3 simulations for $\nu = 7.6 \times 10^{-4}$. The qualitative results are similar to what was observed for the higher Reynolds number on the finer grid, with the shock-capturing scheme missing some of the fine-scale detail and with S2T2 overemphasizing those details. In Figure 4.6 we present the compensated spectrum for these lower Reynolds number runs. At this Reynolds number we do not see a well-developed inertial range, nevertheless the relative behavior of the four schemes tested is almost identical to the higher Reynolds number case. This confirms the scaling relationship derived above, which, in turn, shows that the estimated savings from using a higher-order method is insensitive to

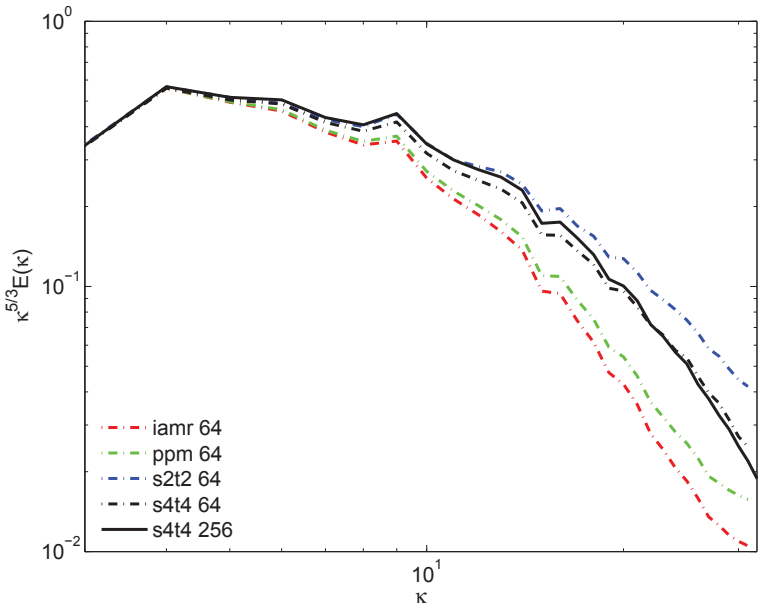

FIG. 4.6. *Compensated spectrum for low Reynolds number case.*

Reynolds number.

5. Summary. We have developed a fourth-order algorithm for the incompressible Navier–Stokes equations based on an auxiliary variable formulation. The methodology uses fourth-order finite volume differencing in space and a fourth-order spectral deferred corrections integration scheme in time. We demonstrated that the method converges at the expected rate for smooth flows. More importantly, we demonstrated that the use of the fourth-order discretization provided a significant advantage for modeling of turbulent flows. In particular, we showed that the fourth-order scheme provided about a factor of two reduction in each direction in the size of the computational mesh needed to resolve a turbulent flow at a given Reynolds number compared to a number of different second-order discretization approaches. This last observation is a key issue in the utility of these types of discretizations for application to more complex low Mach number flow models.

The results presented here open up several avenues of investigation. In terms of the algorithm, variations on the basic discretization, such as choice of quadrature rule and spatial discretization can be investigated. Another promising extension is to embed the iterative temporal integration strategy within adaptive mesh refinement algorithms. The long term goal is to extend the higher-order methods described here to provide the basis for next generation algorithms for zero Mach number flow models in combustion and astrophysics building on existing second-order methods [16, 34].

Appendix. Linear stability of the SDC methods. The linear stability of the temporal integration methods based on a semi-implicit SDC discretization is considered in [27]. Since the particular choice of quadrature nodes and iterations used for the fourth-order time integration method used in this paper is not covered in [27], we include the stability diagrams here for completeness.

In order to study the linear stability of a semi-implicit scheme, one has to first

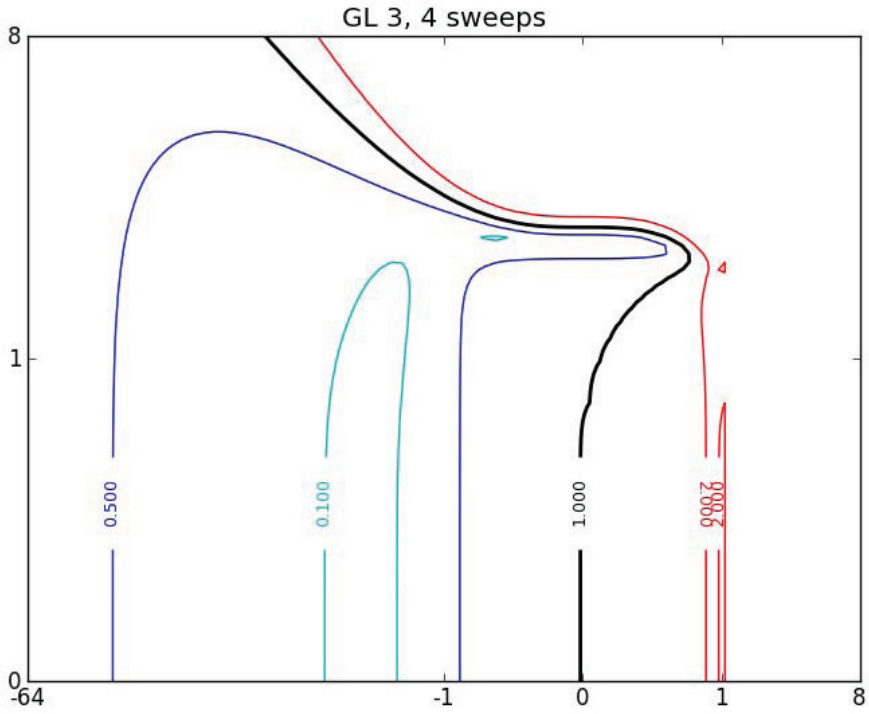

FIG. A.1. *Semi-implicit linear stability diagram for the fourth-order SDC time integration method.*

decide how to split the traditional linear model problem. In the context of PDEs where advection terms are treated explicitly and diffusive terms implicitly (as is the case in the paper), the most logical splitting is

$$y' = \mathbf{I}(\lambda)y + \mathbf{R}(\lambda)y \tag{A.1}$$

for the complex scalar λ. The imaginary term is treated explicitly and real term explicitly. This choice is convenient since one can then produce the traditional stability diagram in the complex plane.

Figure A.1 displays the level curves of the amplification factor in the complex plane. Note that the axes are scaled cubicly as in [27] to provide more detail near the origin. The method is stable for some portion of the imaginary axis, which is desirable for problems with low Reynolds number. The method is also $A(\alpha)$-stable in the semi-implicit sense, but not $L(\alpha)$-stable (again see [27] for a more detailed discussion of $A(\alpha)$ and $L(\alpha)$ stability for semi-implicit methods).

To connect the stability diagrams to a time-step restriction for PDEs, consider a linear advection-diffusion equation

$$u_t = au_x + \nu u_{xx}. \tag{A.2}$$

Using a method of lines discretization and the fourth-order finite volume methods

used in this paper would result in a system of equations with eigenvalues

$$\lambda_k = \mu i \frac{8\sin(\omega_k \Delta x) - \sin(2\omega_k \Delta x)}{6} + \sigma \frac{-15 + 16\cos(\omega_k \Delta x) - \cos(2\omega_k \Delta x)}{6}, \tag{A.3}$$

where $\mu = a/\Delta x$, $\sigma = \nu/\Delta x^2$, and $\omega_k = 2\pi k$ for $k = [0, N-1]$. Linear stability is acheived if $\lambda_k \Delta t$ is within the stability diagram for the particular method. For a three-dimensional problem, the eigenvalues would include the analagous contributions from derivatives in the other dimensions.

REFERENCES

[1] S. Abarbanel, D. Gottlieb, and M. H. Carpenter, *On the removal of boundary errors caused by Runge–Kutta integration of nonlinear partial differential equations*, SIAM J. Sci. Comput., 17 (1996), pp. 777–782.

[2] A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell, and M. L. Welcome, *A conservative adaptive projection method for the variable density incompressible Navier-Stokes equations*, J. Comput. Phys., 142 (1998), pp. 1–46.

[3] I. Alonso-Mallo, *Runge-Kutta methods without order reduction for intial boundary value problems*, Numer. Math., 91 (2002), pp. 577–603.

[4] A. J. Aspden, N. Nikiforakis, S. B. Dalziel, and J. B. Bell, *Analysis of Implicit LES methods*, Commun. Appl. Math. Comput. Sci., 3 (2008), pp. 103–126.

[5] J. B. Bell, P. Colella, and H. M. Glaz, *A second order projection method for the incompressible Navier-Stokes equations*, J. Comput. Phys., 85 (1989), pp. 257–283.

[6] J. B. Bell, M. S. Day, C. A. Rendleman, S. E. Woosley, and M. A. Zingale, *Adaptive low Mach number simulations of nuclear flame microphysics*, J. Comput. Phys., 195 (2004), pp. 677–694.

[7] S. Boscarino, *Error analysis of IMEX Runge–Kutta methods derived from differential-algebraic systems*, SIAM J. Numer. Anal., 45 (2007), pp. 1600–1621.

[8] A. Bourlioux, A. T. Layton, and M. L. Minion, *Higher-order multi-implicit spectral deferred correction methods for problems of reacting flow*, J. Comput. Phys., 189 (2003), pp. 351–376.

[9] E. L. Bouzarth and M. L. Minion, *A multirate integrator for regularized Stokeslets*, J. Comput. Phys., 229 (2010), pp. 4208–4224.

[10] D. L. Brown, R. Cortez, and M. L. Minion, *Accurate projection methods for the incompressible Navier-Stokes equations*, J. Comput. Phys., 168 (2001), pp. 464–499.

[11] M. P. Calvo and C. Palencia, *Avoiding the order reduction of Runge-Kutta methods for linear initial boundary value problems*, Math. Comp., 71 (2002), pp. 1529–1543.

[12] M. H. Carpenter, D. Gottlieb, S. Abarbanel, and W.-S. Don, *The theoretical accuracy of Runge–Kutta time discretization for the initial-boundary value problem: A study of the boundary error*, SIAM J. Sci. Comput., 16 (1995), pp. 1241–1252.

[13] T. L. Clark and R. D. Farley, *Downslope windstorm calculations in two and three spatial dimensions using anelastic interactive grid nesting: A possible mechanism for gustiness*, J. of Atmos. Sci., 41 (1984), pp. 329–350.

[14] P. Colella and P. R. Woodward, *The piecewise parabolic method (ppm) for gas-dynamics simulations*, J. Comput. Phys., 54 (1984), pp. 174–201.

[15] E. M. Constantinescu and A. Sandu, *Extrapolated implicit-explicit time stepping*, SIAM J. Sci. Comput., 31 (2010), pp. 4452–4477.

[16] M. S. Day and J. B. Bell, *Numerical simulation of laminar reacting flows with complex chemistry*, Combust. Theory Modelling, 4 (2000), pp. 535–556.

[17] D. R. Durran, *Improving the anelastic approximation*, J. of Atmos. Sci., 46 (1989), pp. 1453–1461.

[18] W. E and J.-G. Liu, *Gauge method for viscous incompressible flows*, Commun. Math. Sci., 1 (2003), pp. 317–332.

[19] P. Gervasio, *Convergence analysis of high order algebraic fractional step schemes for time-dependent stokes equations*, SIAM J. Numer. Anal., 46 (2008), pp. 1682–1703.

[20] J. L. Guermond, P. Minev, and J. Shen, *An overview of projection methods for incompressible flows*, Comput. Methods. Appl. Mech. Engrg., 195 (2006), pp. 6011–6045.

[21] B. Gustafsson, *The convergence rate for difference approximations to mixed initial boundary value problems*, Math. Comput., 29 (1975), pp. 396–406.

[22] S. Y. Kadioglu, R. Klein, and M. L. Minion, *A fourth-order auxiliary variable projection method for zero-Mach number gas dynamics*, J. Comput. Phys., 227 (2008), pp. 2012–2043.
[23] C. A. Kennedy and M. H. Carpenter, *Additive Runge-Kutta schemes for convection-diffusion-reaction equations*, Appl. Numer. Math., 44 (2003), pp. 139–181.
[24] J. Kim and P. Moin, *Application of a fractional-step method to incompressible Navier-Stokes equations*, J. Comput. Phys., 59 (1985), pp. 308–323.
[25] O. M. Knio, H. N. Najm, and P. S. Wyckoff, *A semi-implicit numerical scheme for reacting flow:* II. *Stiff, operator-split formulation*, J. Comput. Phys., 154 (1999), pp. 428–467.
[26] A. T. Layton and M. L. Minion, *Conservative multi-implicit spectral deferred correction methods for reacting gas dynamics*, J. Comput. Phys., 194 (2004), pp. 697–714.
[27] A. T. Layton and M. L. Minion, *Implications of the choice of quadrature nodes for Picard integral deferred corrections methods for ordinary differential equations*, BIT, 45 (2005), pp. 341–373.
[28] A. Majda and J. A. Sethian, *Derivation and numerical solution of the equations of low Mach number combustion*, Comb. Sci. Tech., 42 (1985), pp. 185–205.
[29] J. J. R. P. A. McMurtry, W.-H. Jou, and R. W. Metcalfe, *Direct numerical simulations of a reacting mixing layer with chemical heat release*, AIAA J., 24 (1986), pp. 962–970.
[30] M. L. Minion, *Higher-order semi-implicit projection methods*, in Numerical Simulations of Incompressible Flows (Half Moon Bay, CA, 2003), M. Hafez, ed., World Scientific Publishing, River Edge, NJ, 2001, pp. 126–140.
[31] M. L. Minion, *Semi-implicit spectral deferred correction methods for ordinary differential equations*, Commun. Math. Sci., 1 (2003), pp. 471–500.
[32] M. L. Minion, *Semi-implicit projection methods for incompressible flow based on spectral deferred corrections*, Appl. Numer. Math., 48 (2004), pp. 369–387.
[33] H. N. Najm, P. S. Wyckoff, and O. M. Knio, *A semi-implicit numerical scheme for reacting flow.* I. *Stiff chemistry*, J. Comput. Phys., 143 (1998), pp. 381–402.
[34] A. Nonaka, A. S. Almgren, J. B. Bell, M. J. Lijewski, C. M. Malone, and M. Zingale, *MAESTRO: An adaptive low Mach number hydrodynamics algorithm for stellar flows*, Astrophysical Journal Supplement, 188 (2010), pp. 358–383.
[35] Y. Ogura and N. A. Phillips, *Scale analysis of deep and shallow convection in the atmosphere*, J. of Atmos. Sci., 19 (1962), pp. 173–179.
[36] R. B. Pember, L. H. Howell, J. B. Bell, P. Colella, W. Y. Crutchfield, W. A. Fiveland, and J. P. Jessee, *An adaptive projection method for unsteady, low-Mach number combustion*, Combust. Sci. Technol., 140 (1998), pp. 123–168.
[37] A. Prohl, *Projection and Quasi-Compressibility Methods for Solving the Incompressible Navier-Stokes Equations*, Adv. Numer. Math., B. G. Teubner, Stuttgart, 1997.
[38] G. Russo and P. Smereka, *Impulse formulation of the Euler equations: General properties and numerical methods*, J. Fluid Mech., 391 (1999), pp. 189–209.
[39] J. Van Kan, *A second-order accurate pressure-correction scheme for viscous incompressible flow*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 870–891.